

Edge Impulse – Voice Recognition

Table of Contents

Introduction to Edge Impulse.....	1
What is Edge Impulse?.....	1
Key Features:.....	1
1. Getting Started with Edge Impulse	2
2. Setting Up Development Board or Mobile Device	2
3. Data Acquisition (Collecting Voice Samples)	3
4. Data Exploration and Split.....	4
5. Create Impulse (Signal + Features + Model)	4
6. Configure MFCC Block	5
7. Training the Neural Network.....	6
8. Model Testing and Evaluation	7
9. Live Classification	8
10. Deployment on Edge Devices.....	8
11. Real-time Testing on Microcontroller or Raspberry Pi	9

Introduction to Edge Impulse

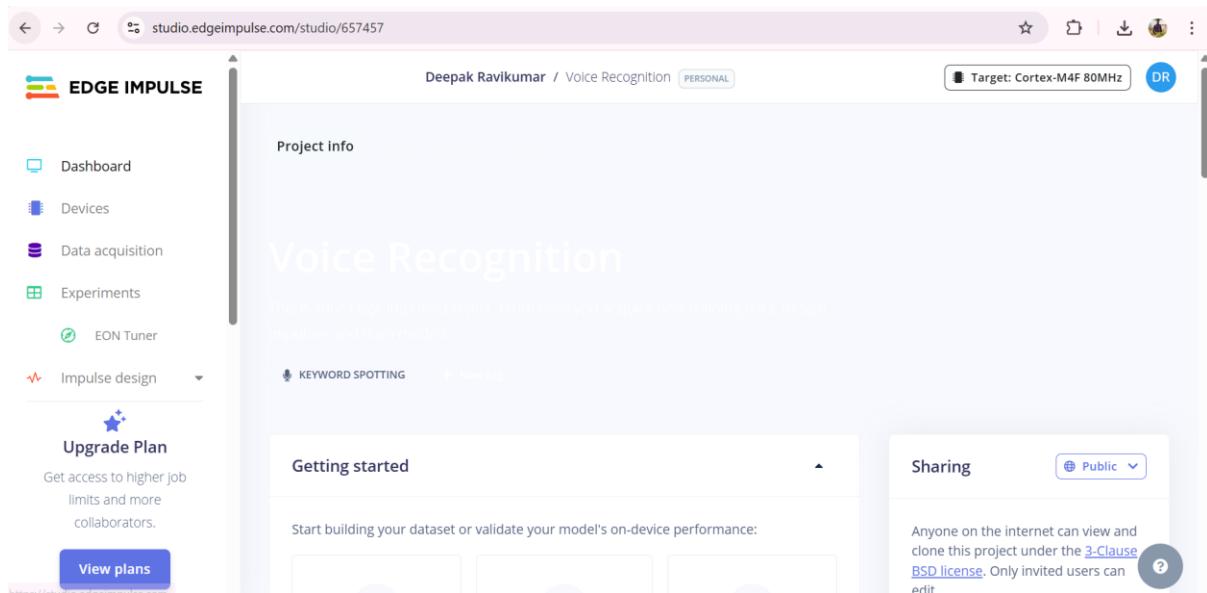
What is Edge Impulse?

Edge Impulse is a development platform for machine learning on edge devices (microcontrollers, sensors, smartphones). It helps developers collect data, train models, and deploy them with low latency and high accuracy.

Key Features:

- Collect real-time audio/sensor data
- Build ML/DL models in the browser
- Edge-optimized deployment
- Supports Arduino, Raspberry Pi, ESP32, STM32, etc.

Use Case Focus: **Voice Recognition** (e.g., recognizing words like "Yes", "No", or commands like "Start", "Stop")



The screenshot shows the Edge Impulse web interface. On the left, there's a sidebar with options like Dashboard, Devices, Data acquisition, Experiments, EON Tuner, and Impulse design. A 'Upgrade Plan' section is also present. The main area is titled 'Voice Recognition' and includes sections for 'Project info', 'Getting started' (with a note to start building a dataset or validate model performance), and 'Sharing' (set to 'Public'). A message at the bottom right states that anyone can view and clone the project under the 3-Clause BSD license.

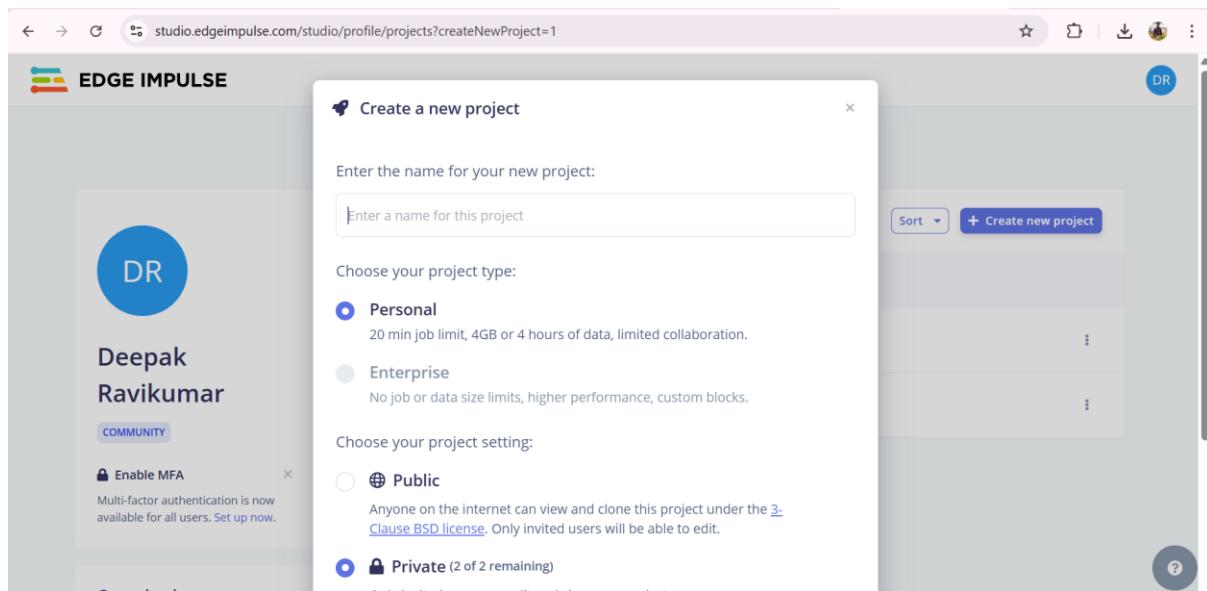
1. Getting Started with Edge Impulse

Objective:

Create an Edge Impulse account and prepare for your voice recognition project.

Steps:

1. Go to <https://edgeimpulse.com>
2. Click **Sign Up** and verify email
3. Log in and click **Create New Project**
 - o Name: "Voice Recognition"
 - o Select type: Classification



The screenshot shows the 'Create a new project' dialog box. It asks for a project name ('Enter a name for this project') and project type ('Choose your project type:'). The 'Personal' option is selected, described as having a 20 min job limit, 4GB or 4 hours of data, and limited collaboration. The 'Enterprise' option is also available. It then asks for a project setting ('Choose your project setting:'), with 'Public' and 'Private' being the choices. The 'Private' option is selected, noted as allowing invited users to edit and view the project.

2. Setting Up Development Board or Mobile Device

Objective:

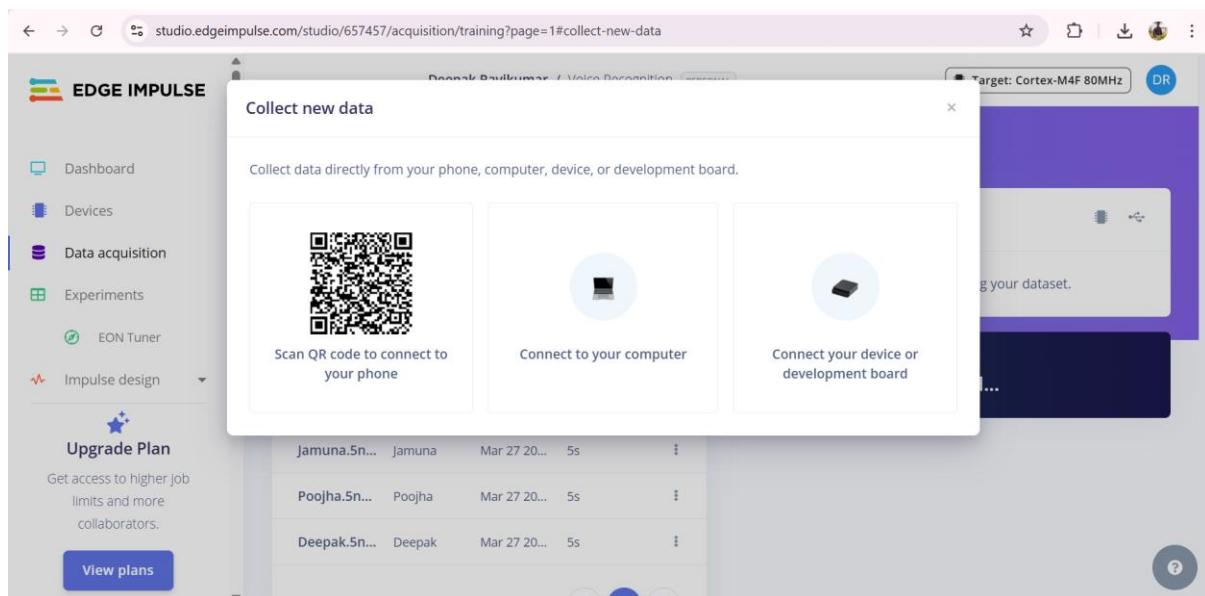
Connect a device for data collection.

Option 1: Using Mobile Phone (Easy)

1. Install Edge Impulse Mobile App
2. Log in with your Edge Impulse account
3. Connect phone to your project via QR Code

Option 2: Using Arduino, ESP32, Raspberry Pi (Advanced)

1. Install Edge Impulse CLI
2. Connect device via USB
3. Run edge-impulse-daemon



3. Data Acquisition (Collecting Voice Samples)

Objective:

Collect voice samples to train the model.

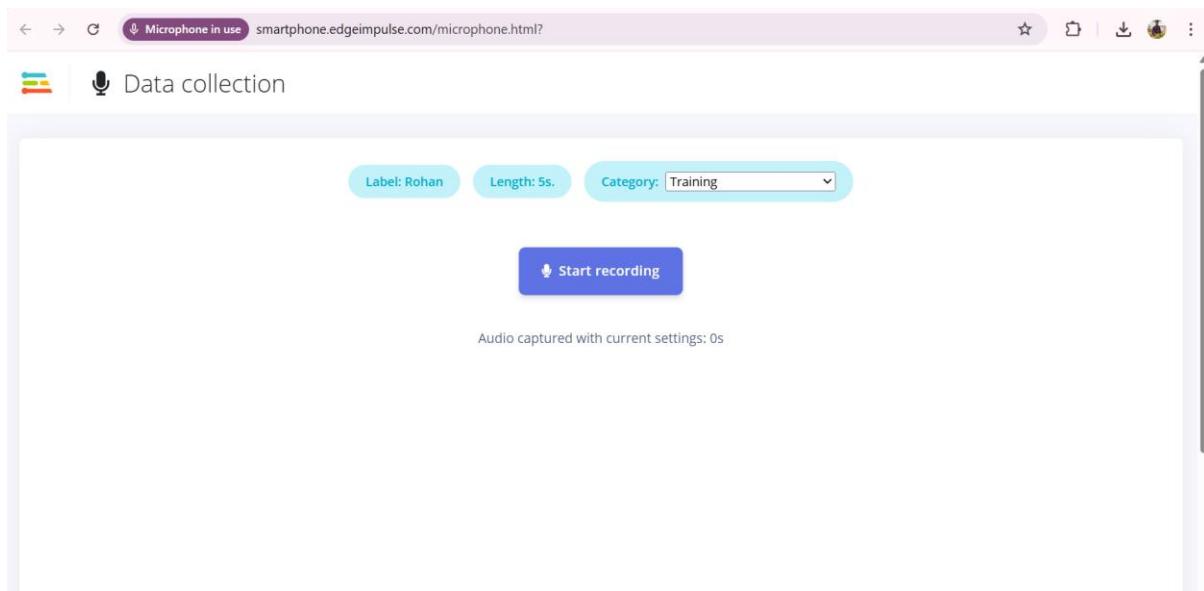
Create Labels:

- Example Labels: yes, no, up, down, noise, unknown

Record Audio:

1. Go to Data Acquisition
2. Set Label and Duration (e.g., 1 second)
3. Use microphone to record voice samples
4. Record ~50 samples per label

Tip: Record in different tones, distances, and environments.



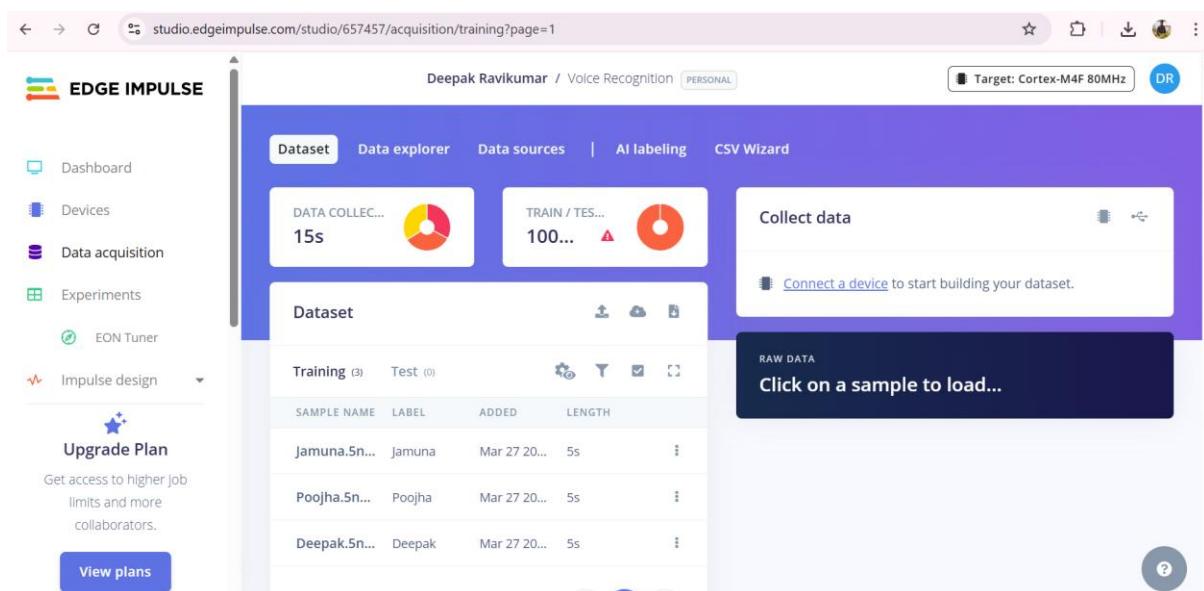
4. Data Exploration and Split

Objective:

Understand and organize collected audio data.

Steps:

1. Go to **Data Acquisition**
2. View raw signal for each sample
3. Automatically split into **Training (80%)** and **Testing (20%)**



SAMPLE NAME	LABEL	ADDED	LENGTH
Jamuna.5n...	Jamuna	Mar 27 20...	5s
Poojha.5n...	Poojha	Mar 27 20...	5s
Deepak.5n...	Deepak	Mar 27 20...	5s

5. Create Impulse (Signal + Features + Model)

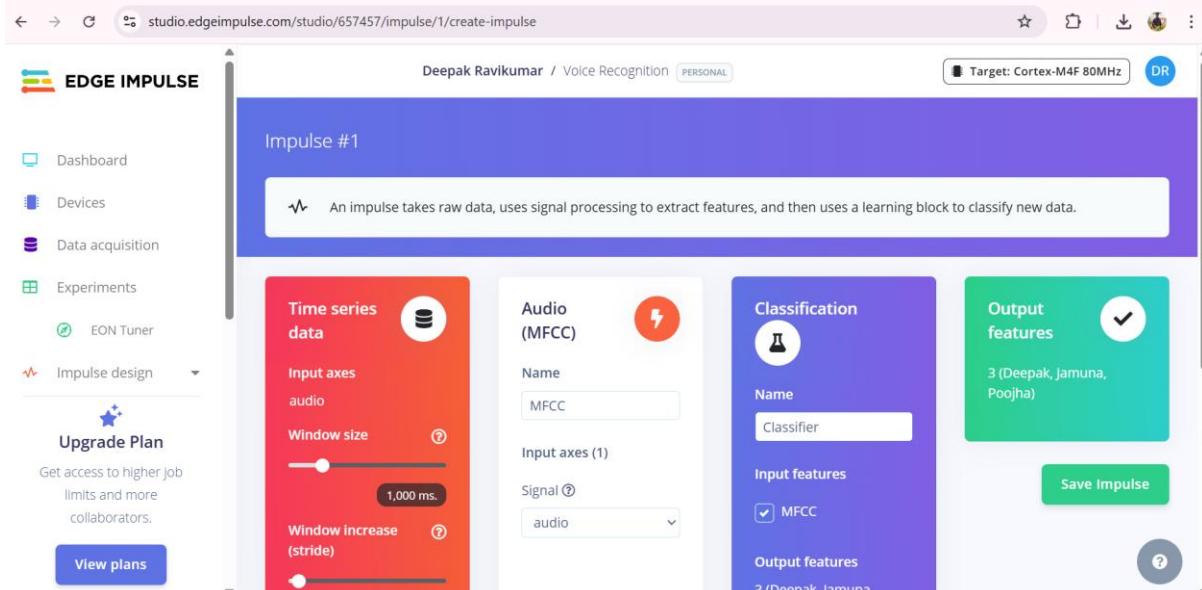
Objective:

Define the ML pipeline for voice classification.

Steps:

1. Go to **Create Impulse**
2. Add a **Processing Block**: MFCC (Mel Frequency Cepstral Coefficient)
3. Add a **Learning Block**: Classification (NN or Keras)

MFCC helps convert voice signals into features.



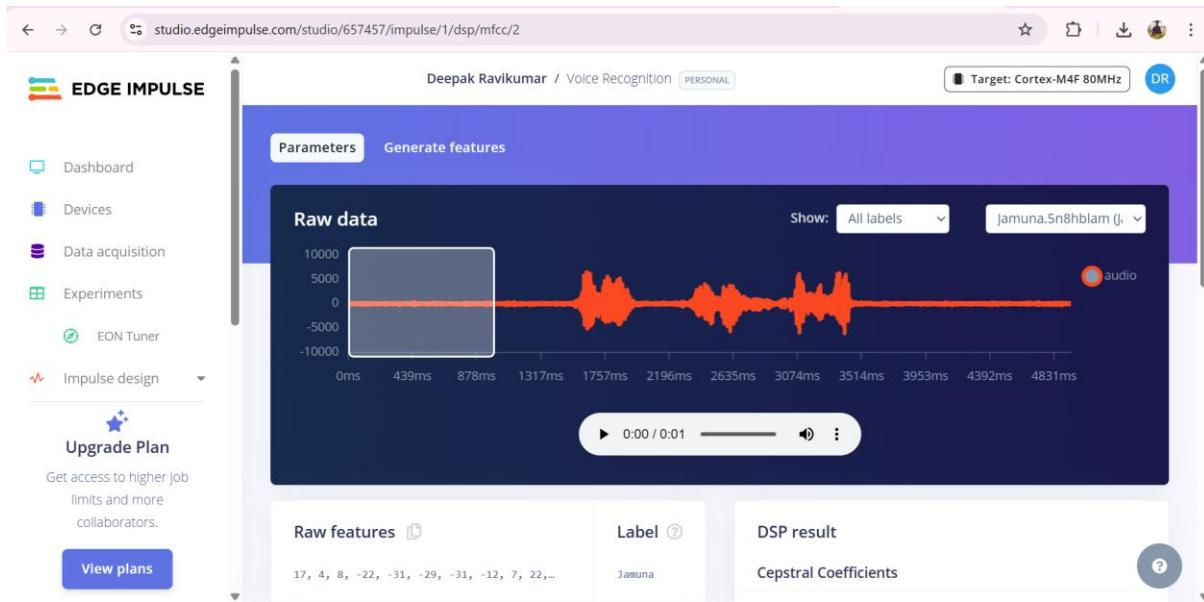
6. Configure MFCC Block

Objective:

Tune audio signal features.

Steps:

1. Go to **MFCC → Parameters**
2. Set:
 - Window size (e.g., 1000ms)
 - Frame length (e.g., 30ms)
 - Frame stride (e.g., 20ms)
3. Click **Save Parameters and Generate Features**



7. Training the Neural Network

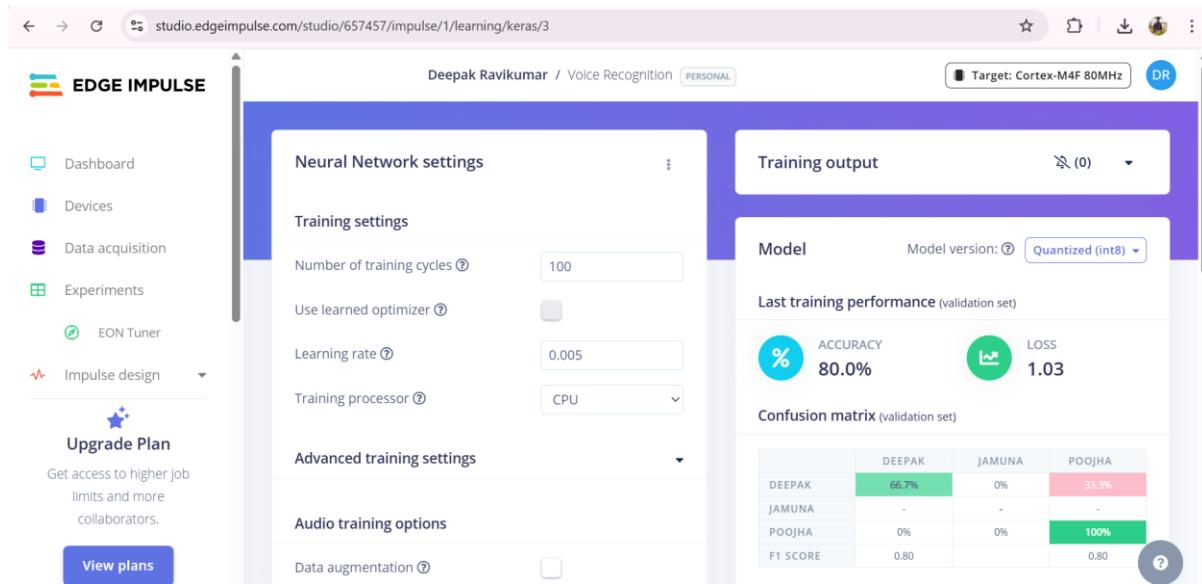
Objective:

Train a model to recognize voice commands.

Steps:

1. Go to **NN Classifier**
2. Set:
 - Learning rate: 0.005
 - Epochs: 40
 - Minimum confidence: 0.65
3. Click **Start Training**

The model uses extracted MFCC features to classify voice commands.



The screenshot shows the Edge Impulse studio interface for a project titled "Voice Recognition". On the left sidebar, under "Impulse design", there is a "Upgrade Plan" section with a "View plans" button. The main area displays "Neural Network settings" and "Training settings". In "Training settings", the "Number of training cycles" is set to 100. The "Model" section shows "Model version: Quantized (int8)". Below it, "Last training performance (validation set)" is listed with Accuracy at 80.0% and Loss at 1.03. A "Confusion matrix (validation set)" table is provided:

	DEEPAK	JAMUNA	POOJHA
DEEPAK	66.7%	0%	33.3%
JAMUNA	-	-	-
POOJHA	0%	0%	100%
F1 SCORE	0.80		0.80

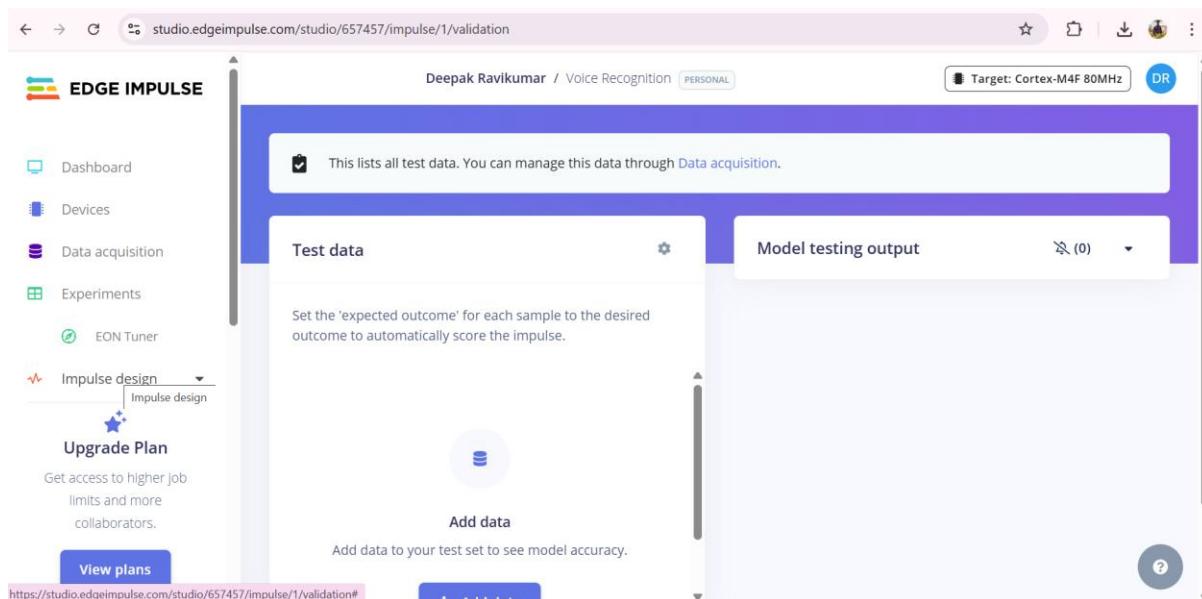
8. Model Testing and Evaluation

Objective:

Test the model accuracy using unseen samples.

Steps:

1. Go to **Model Testing**
2. Click **Classify All**
3. View:
 - o Accuracy
 - o Confusion Matrix
 - o Wrongly classified samples



The screenshot shows the Edge Impulse studio interface for a project titled "Voice Recognition". On the left sidebar, under "Impulse design", there is a "Upgrade Plan" section with a "View plans" button. The main area displays "Test data" and "Model testing output". A message states: "This lists all test data. You can manage this data through Data acquisition." Below it, a note says: "Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse." There is a "Add data" button at the bottom.

9. Live Classification

Objective:

Test voice recognition in real-time using your device.

Steps:

1. Go to **Live Classification**
2. Use mobile mic or board mic
3. Speak a trained word
4. View classification confidence in real-time

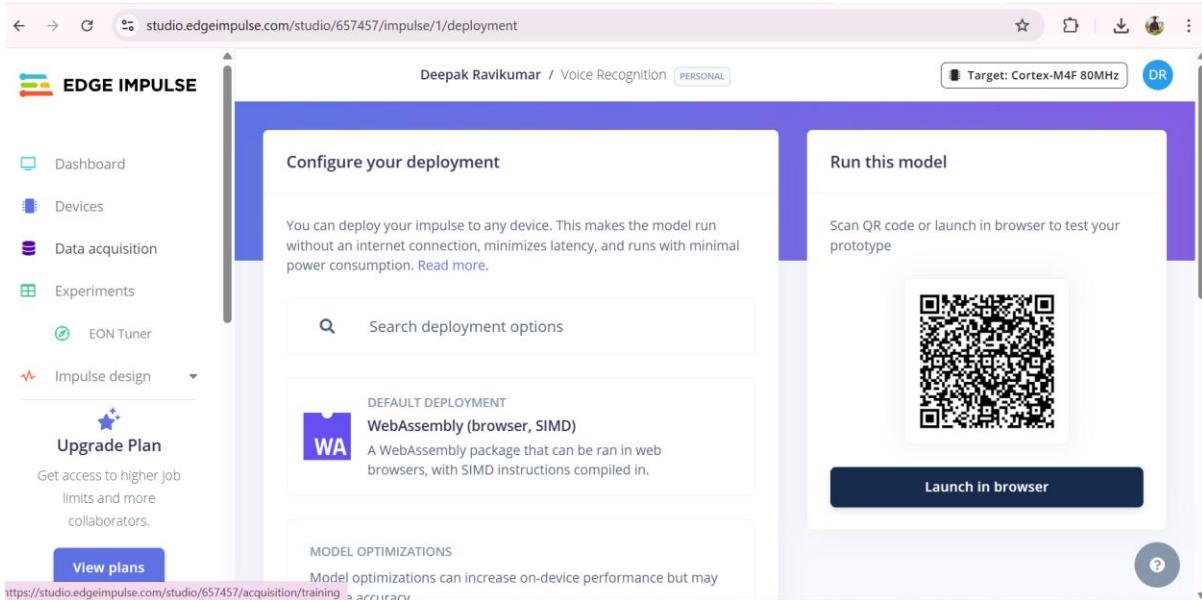
10. Deployment on Edge Devices

Objective:

Deploy model on supported microcontrollers or desktop.

Steps:

1. Go to **Deployment**
2. Select target:
 - o Arduino Library
 - o ESP32 Firmware
 - o Linux x86 (for Raspberry Pi/Desktop)
3. Click **Build**
4. Download and flash onto device



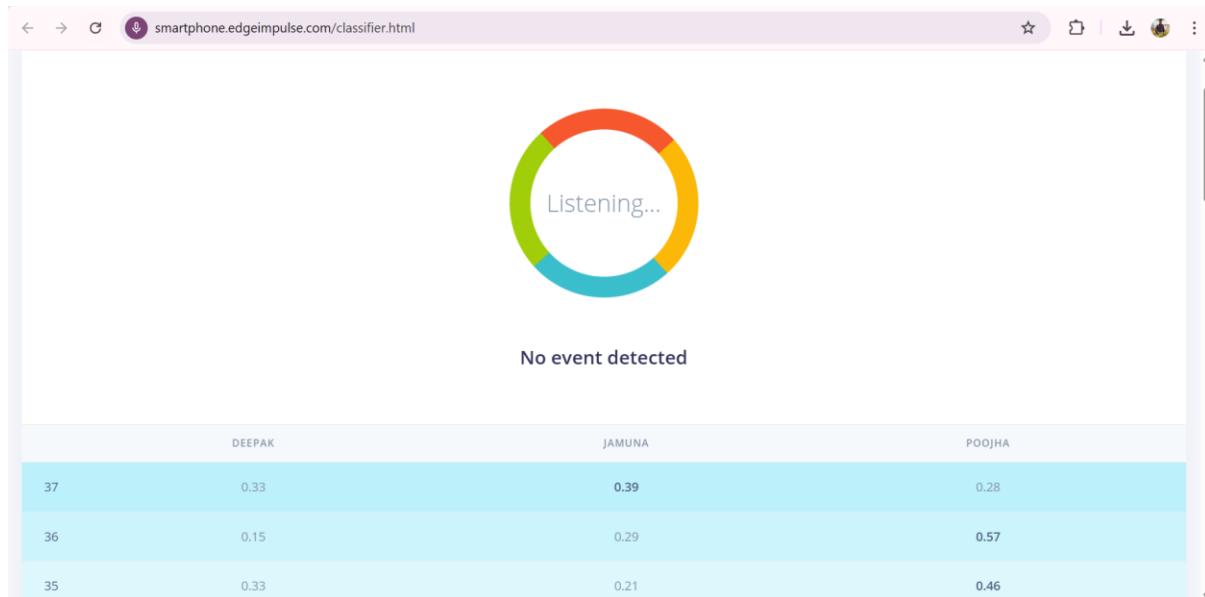
The screenshot shows the Edge Impulse studio interface for deploying a project named "Deepak Ravikumar / Voice Recognition". The deployment target is set to "Cortex-M4F 80MHz". The main area is divided into two sections: "Configure your deployment" and "Run this model".

Configure your deployment: This section allows users to deploy their impulse to any device. It includes a search bar for deployment options and a "DEFAULT DEPLOYMENT" section for "WebAssembly (browser, SIMD)". A note states: "A WebAssembly package that can be ran in web browsers, with SIMD instructions compiled in."

Run this model: This section provides a QR code for testing the prototype. Below the QR code is a "Launch in browser" button.

Left sidebar: Includes links for Dashboard, Devices, Data acquisition, Experiments, EON Tuner, Impulse design, and Upgrade Plan. The Upgrade Plan section mentions getting access to higher job limits and more collaborators, with a "View plans" button.

Bottom footer: Shows the URL <https://studio.edgeimpulse.com/studio/657457/acquisition/training>.



11. Real-time Testing on Microcontroller or Raspberry Pi

Objective:

Run the voice recognition model on hardware.

Example:

- Connect a USB mic to Raspberry Pi
- Run:
`edge-impulse-linux-runner`
- Say commands and watch predictions on terminal.